

CLAIMS

We claim:

1. A system, comprising:
5
one or more processors;
memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement:
10
one or more applications configured to initiate one or more transactions, wherein each of the one or more transactions comprises requests to access one or more data sources; and
15
a transaction manager configured to manage the one or more transactions initiated by the one or more applications, wherein the transaction manager is configured to pause the one or more transactions in response to a pause request and to resume the one or more transactions in response to a resume request, wherein while
20
paused, the transaction manager does not allow the one or more transactions to complete.
2. The system as recited in claim 1, wherein the transaction manager is
25
configured to change the state of the one or more transactions.
3. The system as recited in claim 2, wherein the transaction manager is
configured to request permission to change the state of the one or more transactions.
4. The system as recited in claim 3, wherein the transaction manager is
30
configured to wait for permission to change the state of the one or more transactions.

5. The system as recited in claim 1, wherein the transaction manager is configured to prohibit the change of state of the one or more transactions while the transaction manager is paused.

5 6. The system as recited in claim 1, wherein the transaction manager is configured to support the execution of the one or more transactions within a current state while the transaction manager is paused.

7. The system as recited in claim 1, wherein the system is configured to
10 continue the execution of the one or more applications while the transaction manager is paused, except for the changing of transaction states.

8. The system as recited in claim 1, wherein while the transaction manager is paused, the system is configured to perform operations on one or more individual
15 transactions.

9. The system as recited in claim 8, wherein the operations comprise one or more from the following: rollback, abort, partial rollback, add/remove participant, and commit.
20

10. The system as recited in claim 1, wherein one or more of the transactions are local transactions.

11. The system as recited in claim 10, wherein while paused, the transaction
25 manager is not allowed to change the state of the one or more transactions to the committing state.

12. The system as recited in claim 1, wherein one or more of the transactions are global transactions.
30

13. The system as recited in claim 12, wherein while paused, the transaction manager is not allowed to change the state of the one or more transactions to the preparing state nor to the committing state.

5 14. A system, comprising a plurality of computer systems coupled by one or more networks, wherein the computer systems comprise:

one or more processors; and

10 memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement one or more application servers comprising:

one or more applications configured to initiate one or more transactions,
15 wherein each of the one or more transactions comprises requests to access one or more data sources; and

one or more transaction managers configured to manage the one or more transactions initiated by the one or more applications, wherein one
20 of the transaction managers is configured to pause a corresponding one or more transactions in response to a pause request and to resume the corresponding one or more transactions in response to a resume request, wherein while paused, the transaction manager does not allow the corresponding one or more transactions to
25 complete.

15. A method, comprising:

generating a request to pause a transaction manager;

pausing the transaction manager in response to said request, wherein while the transaction manager is paused, transactions managed by the transaction manager are prohibited from completing;

5 generating a request to resume the transaction manager; and

resuming the transaction manager in response to said request, wherein when the transaction manager is resumed, transactions managed by the transaction manager are allowed to complete.

10

16. The method as recited in claim 15, wherein said pausing comprises prohibiting the transaction manager from changing the state of the one or more transactions, wherein the transaction manager attempts to perform a state change on a transaction in response to input to the transaction manager.

15

17. The method as recited in claim 16, wherein the input comprises notification that an application has initiated a transaction.

20

18. The method as recited in claim 17, wherein said state change comprises a change from a nonexistent state to an active state.

19. The method as recited in claim 16, wherein the input comprises notification that an application has executed a commit transaction command.

25

20. The method as recited in claim 19, wherein said state change comprises a change from an active state to a preparing state.

21. The method as recited in claim 16, wherein the input comprises notification that all participants are prepared commit the transaction.

30

22. The method as recited in claim 21, wherein said state change comprises a change from a preparing state to a committing state.

23. The method as recited in claim 16, wherein the input comprises
5 notification that all participants have committed the transaction.

24. The method as recited in claim 23, wherein said state change comprises a change from a committing state to a nonexistent state.

10 25. The method as recited in claim 16, further comprising, while the transaction manager is paused, continuing to support the execution of the one or more applications, except for the changing of transaction states.

26. A carrier medium comprising program instructions, wherein the program
15 instructions are computer-executable to:

generate a request to pause a transaction manager;

20 pause the transaction manager in response to said request, wherein while the transaction manager is paused, transactions managed by the transaction manager are prohibited from completing;

generate a request to resume the transaction manager; and

25 resume the transaction manager in response to said request, wherein when the transaction manager is resumed, transactions managed by the transaction manager are allowed to complete.

27. The carrier medium as recited in claim 26, wherein the program
30 instructions are computer-executable to pause the transaction manager by prohibiting the transaction manager from changing the state of the one or more transactions, wherein the

transaction manager attempts to perform a state change on a transaction in response to input to the transaction manager.

28. The carrier medium as recited in claim 27, wherein the input comprises
5 notification that an application has initiated a transaction.

29. The carrier medium as recited in claim 28, wherein said state change comprises a change from a nonexistent state to an active state.

10 30. The carrier medium as recited in claim 27, wherein the input comprises notification that an application has executed a commit transaction command.

31. The carrier medium as recited in claim 30, wherein said state change comprises a change from an active state to a preparing state.

15 32. The carrier medium as recited in claim 27, wherein the input comprises notification that all participants are prepared commit the transaction.

33. The carrier medium as recited in claim 32, wherein said state change
20 comprises a change from a preparing state to a committing state.

34. The carrier medium as recited in claim 27, wherein the input comprises notification that all participants have committed the transaction.

25 35. The carrier medium as recited in claim 34, wherein said state change comprises a change from a committing state to a nonexistent state.

36. The carrier medium as recited in claim 27, wherein the program
instructions are computer-executable to, while the transaction manager is paused,
30 continue to support the execution of the one or more applications, except for the changing of transaction states.